Neuroscience

Homework3 Report

Sana Aminnaji 98104722

1. Two-layer neural network in hand digit problem

1.1 Loading the dataset

100 random data

1.2 Separating train data and test data

In this section, we have defined four different variables as below:

1. X_train: a 3000x400 matrix, each 300 rows are for one number.
2. X_test: a 2000x400 matrix, each 200 rows are for one number.
3. y_train: a 3000x1 matrix, each element shows that the same row in the X_train matrix is identifying which number.
4. y_test: a 2000x1 matrix, each element shows that the same row in the X_test matrix is identifying which number.

1.3 The structure of the neural network

Random initialization:

If we set the initial value of all the weights 0.0 then, the equations of the learning algorithm would fail to make any changes to the network weights, and the model will be stuck. Perhaps the only property known with complete certainty is that the initial parameters need to “break symmetry” between different units. If two hidden units with the same activation function are both connected to the same inputs, then these units must have different initial parameters. If they have the same initial parameters, then a deterministic learning algorithm applied to a deterministic cost and model will constantly update both of these units in the same way.

The weight matrix size:

Each neuron has two different type of values that should be defined:

1. weight
2. threshold value

Which they can be combined into one group and beside the n values for the weights of inputs, we can define another single value for each neuron in order to define the threshold value.

1.4 The theoretical algorithm for computing cost function and its derivation

$$sigmoid(x) = \frac{1}{1+e^{-x}}$$

$$\frac{d}{dx}\big(sigmoid(x)\big) = \big(sigmoid(x)\big) \times \big(1 - sigmoid(x)\big) = \frac{e^{-x}}{(1+e^{-x})^2}$$

$$\frac{d}{d\Theta_{rs}} J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}[y_k^{(i)}\frac{d}{d\Theta_{rs}}((\log(h_\theta(x^{(i)}))_k)) + \left(1-y_k^{(i)}\right)\frac{d}{d\Theta_{rs}}\left((\log(1-h_\theta(x^{(i)}))_k)\right)\Big] + \frac{\lambda}{m}\Theta_{rs}$$

$$\begin{aligned}\frac{d}{d\Theta_{rs}}\big(\log(h_\theta(x))\big) &= \frac{d}{d\Theta_{rs}}\big(\log(sigmoid(\theta^T x))\big) = \frac{1}{sigmoid(\theta^T x)}\frac{d}{d\Theta_{rs}}\big(sigmoid(\theta^T x)\big) \\ &= \frac{1}{sigmoid(\theta^T x)}\big(sigmoid(\theta^T x)\big)\big(1 - sigmoid(\theta^T x)\big)\frac{d}{d\Theta_{rs}}(\theta^T x) \\ &= \big(1 - sigmoid(\theta^T x)\big)\frac{d}{d\Theta_{rs}}(\theta^T x)\end{aligned}$$

$$\begin{aligned}\frac{d}{d\Theta_{rs}}\big(\log(1-h_\theta(x))\big) &= \frac{d}{d\Theta_{rs}}\big(\log(1 - sigmoid(\theta^T x))\big) \\ &= \frac{1}{1 - sigmoid(\theta^T x)}\frac{d}{d\Theta_{rs}}\big(1 - sigmoid(\theta^T x)\big) \\ &= \frac{-1}{1 - sigmoid(\theta^T x)}\big(sigmoid(\theta^T x)\big)\big(1 - sigmoid(\theta^T x)\big)\frac{d}{d\Theta_{rs}}(\theta^T x) \\ &= -sigmoid(\theta^T x)\frac{d}{d\Theta_{rs}}(\theta^T x)\end{aligned}$$

$$\begin{aligned}\rightarrow \sum_{i=1}^{m}\sum_{k=1}^{K}[y_k^{(i)}\frac{d}{d\Theta_{rs}}((\log(h_\theta(x^{(i)}))_k)) + \left(1-y_k^{(i)}\right)\frac{d}{d\Theta_{rs}}\left((\log(1-h_\theta(x^{(i)}))_k)\right)\Big] \\ = \sum\sum\frac{d}{d\Theta_{rs}}(\theta^T x)(y - sigmoid(\theta^T x))\end{aligned}$$

1.5 Neural network using back propagation

The defined functions in this part:

One_hot: inverts the m x 1 array of output to a m x k array.

Sigmoid: calculates the sigmoid of a matrix.

Sigmoidgrad: calculates the sigmoid gradient of a matrix.

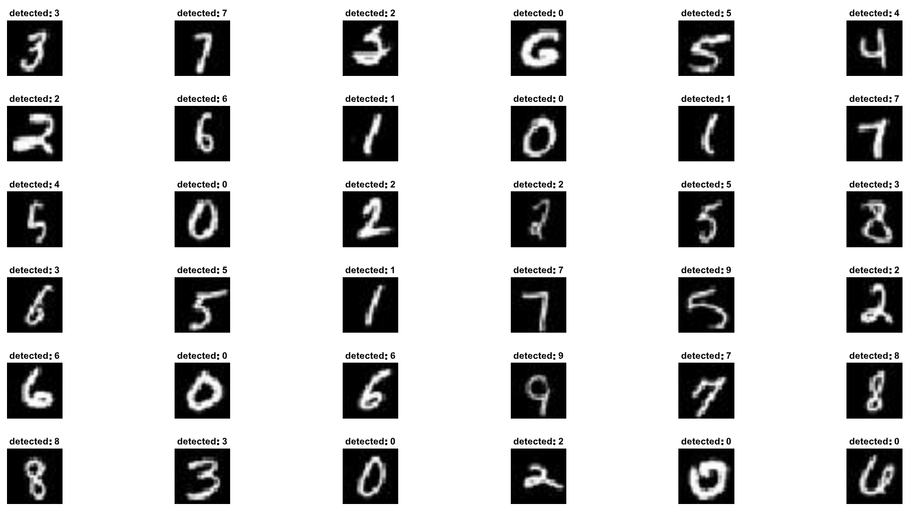
Initial: creates a fix-sized matrix with random values.

CostFunction: calculates the cost value and the gradient matrix.

1.6 Processing neural network

The final weight matrixes named: Theta1 and Theta2.

1.7 The final layer result



1.8 The middle layer result and computing the precision

The variable "hiddenlayer_output" defines the results of the hidden layer, and the variable accuracy is the calculated accuracy of our function.